student

Function Student()
this.stud_id=" " this.grade=[ ] this.div=" "

marks

Function Marks()
Student.call(this) this.eng=" " this.math=" " this.sci=" " this.ss=" "
Marks.prototype = Object.create (Student.prototype); Marks.prototype.constructor = Marks;

calculate

Function Calculate()
Marks.call(this) this.sum=" " this.avg=" " this.pass=[ ] this.fail=[ ] this.total=[ ] this.overall_pass=" " this.overall_fail=" " this.total=" "
Calculate.prototype = Object.create (Marks.prototype); Calculate.prototype.constructor = Calculate;

display

Function Display_array()
Calculate.call(this) this.overall_pp=" "
Display.prototype = Object.create (Calculate.prototype); Display.prototype.constructor = Display;

Hierarchy

```
┌─────────────┐
│  Students   │
└─────────────┘
       │
       ▼            has
┌─────────────┐
│    Marks    │
└─────────────┘
       │
       ▼            has
┌─────────────┐
│  Calculate  │
└─────────────┘
       │
       ▼            has
┌─────────────┐
│   Display   │
└─────────────┘
```